003594.P016 *Patent*

UNITED STATES PATENT APPLICATION

FOR

FORMATTING COMPUTER GENERATED DOCUMENTS FOR OUTPUT

INVENTOR:

ROGER L. BUIS
REINHARD H. HOHENSEE
SUSAN C. MCELRAFTH
ALAN L. MIDDENDORF
JAMSIE R. TREPPENDAHL

Prepared By:

Blakely, Sokoloff, Taylor & Zafman
7887 E. Belleview Avenue, Suite 1290
Denver, CO 80111-6098

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EL328715663US
Date of Deposit: December 30, 1999
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231

Cindy Bennett
(Typed or printed name of person mailing paper or fee)

Cindy Bennett
(Signature of person mailing paper or fee)

12/30/99
(Date signed)

FIELD OF THE INVENTION

The invention relates generally to computer formatting of documents for output. More specifically, the invention relates to a method of formatting a data stream, where each data record contains a layout identifier, according to a set of layout parameters.

BACKGROUND OF THE INVENTION

Invoices, financial statements, and telephone records are examples of documents that typically contain information or data specific to a specific person or entity. The information or data contained in these documents is generally generated by a computer running a data extraction program designed to mine the data from a database. Periodically, a statement may be generated, printed and sent to an account holder for his/her records and review.

During early days of computer–generated documents, documents containing data specific to a certain account holder were printed on impact printers that were limited to producing line output of EBCDIC or ASCII characters. Accordingly, a data stream was formatted by a data extraction program such that each data record represented one line on the resulting printed document. Figure 1a is an example of line data for a customer invoice produced by a data extraction program.

The formatted data was often printed on preproduced forms containing headers, footers, logos, lines, columns and rows. Figure 1b is an example of the customer invoice resulting from the line data in figure 1a.

As printer technology advanced, so did the demand for more sophisticated printed output. The advent of electrostatic and inkjet printers eliminated the need to have preprinted forms containing graphics, logos or any other standard data that is fixed for all documents of a certain type, since the printers were capable of printing both the standard data and the variable data concurrently.

Formatting programs were developed to reformat and remap line data received from the data extraction program and to merge the line data with the standard information on a document. However, because these formatting programs rely on line data received from legacy data extraction programs that did not have a mechanism to facilitate complex formatting the line data records, the formatting programs typically rely upon conditional processing to identify the occurrence of specific text strings or combinations thereof at specific locations in the line data to remap the line data based on these occurrences. One of the primary limitations of this remapping method is that it relies on the ability of the formatting program to distinguish between types of data based on the content of the line data. If two different types of data are similar in form, location and/or structure within the line data, the formatter may not be able to distinguish between. As a result, these formatting programs are limited with regard to the amount and complexity of the reformatting possible.

Another method used to format line data was to use the skip to channel controls to place line data sequentially on a printed page. Typically, a number from 1 to 12 was placed in the first two columns of a plurality of line data records. At first, the numbers were provided to indicate to an impact printer to skip a certain number of lines on a page before printing the succeeding data record for the purpose of saving printing time by not requiring the relatively slow impact printer to scroll across what would end up as blank lines on a finished document. As print technology advanced, the skip to channel numbers were used by a formatter to define placement of a line data record at a fixed vertical position on a page. For instance, a “1” in the first column of a line data record might indicate to skip to the next page before printing the line data following the number “1,” or a “2” might indicate that the line data following it be printed three lines below the top of the page. The process is limiting in that the skip to channel control only controls placement of a line record relative to the top of the page.

Another method of obtaining complex formatted printed documents has been to integrate the function of producing complex document layouts within the data extraction program. Essentially, the data extraction program formats the data record into a desired format rather than as line data. This method allows the creation of complex document formats but it can be expensive. For instance, a tailored data extraction program might be required for each document type, and any changes to a document's format requires that the data extraction program be reprogrammed. Additionally, any programmer tasked to program or reprogram the data extraction program to implement a new document format would potentially have access to sensitive data stored in the database; thereby, creating a potential security risk.

In view of the foregoing, it is desirable to have a method for formatting a document containing variable data that offers a high degree of flexibility in the arrangement of print data as to take full advantage of the capabilities of electrostatic and inkjet printers. It is also desirable that the method minimize the expense of generating formatted documents and that a format for a specific document be capable of easy modification without requiring access to the data extraction program.

SUMMARY OF THE INVENTION

A method and apparatus are disclosed for formatting a data stream, which has a layout identifier for each data record contained therein, according to a set of layout parameters. Embodiments of the current invention include a number of elements and limitations with no single element or limitation contemplated as being critical. Each claim, individually, recites an aspect of the invention in its entirety. Embodiments of the present invention include methods, computer-readable mediums containing instructions for performing a method, and computer systems. The embodiments described herein embody various aspects and configurations of the invention.

According to one embodiment of the invention, a method of formatting documents includes assigning a layout identifier to each data record retrieved from a database based on the type of data contained within the data record and how the data is to be formatted in the document. Specific layout parameters are assigned to each layout identifier that specify the placement of an associated print data record within a document to form, collectively, a set of layout parameters. Next, the set of layout parameters are applied to a data stream and each of the data records contained therein to create a formatted document. In different embodiments of the system the various elements of the system may be integrated into a single software program, co-resident on a single computer system, or distributed across various computer systems. It is contemplated that one or more of the various elements of the invention could be embodied as hardware or software.

In another embodiment, a formatter receives a data stream in which each data record therein is associated with a layout identifier. The formatter also receives a set of layout parameters in which formatting instructions are provided corresponding to each layout identifier. The layout parameters corresponding to a particular layout identifier are applied to each data record associated with the particular layout identifier. The result of the formatting process is a document that may be sent to an output device for display or printing.

In another embodiment, a data extraction program retrieves data from a database. The data is arranged into data records to form collectively a data stream. A layout identifier is associated with each data record, where the layout identifier identifies certain characteristics of the data record to which it is associated. The one or more data records and the associated layout identifiers are sent to a formatting program.

In another embodiment, a layout program specifies layout parameters for each layout identifier that is or is likely to be present in a particular type of document in the form of layout parameters. The specific layout parameters for each layout identifier are combined to form a set of layout parameters. The set of layout parameters corresponding to a certain document type are sent to a formatter.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of the present invention with particularity. The invention is illustrated by way of example, and not by way of limitation, from the following detailed description taken in conjunction with the accompanying drawings of which:

Figure 1a is an example of line data produced by a data extraction program in the prior art;

Figure 1b is an example of printed output produced using the line data from figure 1a under the prior art;

Figure 2 is an illustration of an exemplary computer system upon which embodiments of the present invention may be implemented;

Figure 3 is a block diagram illustrating a computing environment in which embodiment of the invention may be practiced;

Figure 4 is a flow diagram for an exemplary data extraction program;

Figure 5 is a flow diagram for an exemplary layout program;

Figure 6 is a flow diagram for an exemplary formatter;

Figure 7 is an example a data stream of line data records with layout identifiers for a bank statement;

Figure 8 is an example of a bank statement formatted according to an embodiment of the invention;

Figures 9 & 10 are layout parameter scripts generated by a layout program corresponding to particular layout identifiers for the bank statement example;

Figure 11 is a block diagram of a computing environment that might be used to produce the example Bank Statement.

DETAILED DESCRIPTION OF THE PREFFERED EMBODIMENT

A method and apparatus are disclosed for formatting a data stream, which has a layout identifier for each data record contained therein, according to a set of layout parameters. Embodiments of the invention provide a flexible process for formatting data records extracted from a database for forwarding to an output device such as a display or printer in which visually pleasing and potentially complex documents may be produced. An additional advantage of the invention is that it provides for quick and inexpensive modification of a document format without having to reprogram.

The invention is not limited to a single computing environment. Moreover, the architecture and functionality of the invention as taught herein and as would be understood by one skilled in the art is extensible to a number of computing environments and embodiments in keeping with the scope and spirit of the invention. The invention provides for various methods, computer-readable medium containing computer-executable instructions, and apparatus. With this in mind, the embodiments discussed herein should not be taken as limiting the scope of the invention; rather, the invention contemplates all embodiments as may come within the scope of the appended claims.

Terminology

Format Region – An area within a page of a document (e.g., a page header, a page footer, a body, and a group header). All data and information contained within the format region follow the layout conventions assigned to that format region.

Layout identifier – A name or handle associated with a data record in a data stream. Typically, a layout identifier acts to identify the type of data within an associated data record and allows the formatter to associate the data record with particular formatting instructions.

Layout Parameters – a series of formatting instructions for data records contained in a data stream and for format areas. Typically, layout parameters will specify how a particular data record will be formatted in a document based on the layout identifier associated with it. Additionally, there may be layout parameters that specify the location and size of various format areas within a document. A set of layout parameters contains layout parameters for each type of layout identifier present in a related data stream, as well as, formatting instructions for any format regions defined for the resulting formatted document.

An Exemplary Computer System

Figure 2 provides an exemplary computer system in which the features of the present invention may be implemented.

Computer system 200 comprises a bus or other communication means 201 for communicating information, and a processing means such as processor 202 coupled with bus 201 for processing information. Computer system 200 further comprises a random access memory (RAM) or other dynamic storage device 204 (referred to as main memory), coupled to bus 201 for storing information and instructions to be executed by processor 202. Main memory 204 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 202. Computer system 200 also comprises a read only memory (ROM) and/or other static storage device 206 coupled to bus 201 for storing static information and instructions for processor 202.

A data storage device 207 such as a magnetic disk or optical disc and its corresponding drive may also be coupled to computer system 200 for storing information and instructions. Computer system 200 can also be coupled via bus 201 to a display device 221, such as a cathode ray tube (CRT) or Liquid Crystal Display (LCD), for displaying information to a computer user. Typically, an alphanumeric input device 222, including alphanumeric and other keys, is coupled to bus 201 for communicating information and/or command selections to processor 202. Another type of user input device is a cursor controller 223, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 202 and for controlling cursor movement on display 221.

A communication device 224 may also be coupled to bus 201 for accessing remote servers, or clients. The communication device 224 may include a modem, a network interface card, or other well-known interface devices, such as those used for coupling to an Ethernet, token ring, or other types of networks. Through this communication device 224, the computer system 200 may be coupled to a number of servers via a conventional network infrastructure, such as a company's Intranet and/or the Internet, for example.

The invention as claimed in various embodiments may be practiced on a single computer system or the various elements of the invention may be distributed over several computer systems.

An Exemplary Method of Formatting Documents

Figure 3 illustrates an exemplary computing environment in which one embodiment of the invention may be practiced. This example shows various elements involved in the process of creating a formatted document 355 of one or more pages that contains formatted data as retrieved from a database 310. Although the embodiment discussed herein references a printed document, it is to be understood that the presentation format of the document produced using the invention is not critical. A document could encompass any conceivable format that is produced by an output device, including information displayed on a computer monitor.

Whenever a formatted document 355, consisting of data extracted and derived from a database 310, is desired, a data extraction program 320 is run to retrieve data 315, perform calculations as necessary on the data 315, create a data stream 325 comprising data records, and associate a layout identifier with each line data record. Next, the line data stream 325 is sent to a formatter 340.

A set of layout parameters 335 are created by a layout program 330 that specify rules and layout information relative to the formatter 340 to layout the formatted document 355. The layout program 330 sends the set of layout parameters 335 to the formatter 340. It is envisioned in certain embodiments that the layout program 330 could be run on a computer using a Windows O/S, where the layout program 330 provides a GUI interface to which the user can generate a facsimile of a desired formatted document 355 based on the generated set of layout parameters 335 for review prior to sending the layout parameters 335 to the formatter.

When the formatter 340 receives the data stream 325, the formatter 340 associates the data stream 325 with a set of layout parameters 335 and applies the rules and information contained within the set of layout parameters 335 to the data stream 325 to create a document data file 345 that is sent to an output device 350. Finally, an output device 350, such as a printer or a monitor, produces the formatted document 355.

An Exemplary Data Extraction Program

Figure 4 is a flow diagram showing the steps performed by an exemplary data extraction program 320. First in step 410, the data extraction program 320 requests and receives raw data 315 from the database 310. As necessary in step 420, the data extraction program 320 performs calculations on the data 315 creating new data or information to be included in the formatted document 355. For instance, the data extraction program 320 might determine an average from a set of numbers contained in the data 315. The data 315 and any new data are arranged into data stream

325 comprising a plurality of data records as shown in step 430. In step 440, a layout identifier is then associated with each data record based on the type and nature of the information contained within the particular data record. Finally in step 450, the data stream 325 and the associated layout identifiers are sent to a formatter 340.

It is envisioned that the data extraction program 320 could be implemented in software or hardware in a number of different embodiments. For instance, the data extraction program 320 could be resident on a computer system 200 containing the database 310, or it could reside on another computer system 200 that interacts with the computer system 200 containing the database 310 over a network. Alternatively, the data extraction program 320 could be a routine of a database program. Furthermore, the steps and the order of steps in figure 4 are merely exemplary and as such, it is envisioned that a data extraction program 320 may have more or less steps than illustrated and that the steps may be performed in a different order than illustrated.

An Exemplary Layout Program

Figure 5 is a flow diagram showing the steps performed by an exemplary layout program 330 in the creation of a set of layout parameters 335.

The first step 510 in generating a set of layout parameters 335 is to setup and define various format regions that will appear on a page within a document. Essentially, a format region defines an area on a page where all data contained within the format region follow the layout conventions assigned to that format region. For some embodiments of the invention, format regions are not specified.

A typical document page might include a page header, a body, and a page trailer. There may also be group headers that are printed on a page preceding certain groups of data. Additionally, other format regions are contemplated. Parameters are set for each format region in step 520, such as where the format region will appear on a formatted page, and whether its location on the page is relative to other format regions. Definitions of typical format regions are provided below:

BODY

The majority of the data retrieved from a data extraction program 320 as data records are placed in this format region on a formatted document 355. Typically, data contained herein is printed line by line.

PAGEHEADER

This format region specifies a header that is to be printed on each new page within a formatted document 355. The baseline position of this format region is normally in the top margin, but it can be positioned anywhere relative to the page origin. Typically, the header contains standard information such as the name, address and logo of the entity producing the document (e.g. a bank's name and address on a bank statement). It may also contain the page header contains information regarding a customer or account holder such as a name, an address, a phone number, and account number.

GRPHEADER

This format region specifies a header that is to be printed at the beginning of a group of data. If all of the data within a group of data cannot be formatted to fit on to a page within a formatted document 355, the group header will be reprinted on the next page after the page header, as applicable, and on each subsequent page within the formatted document 355 until all the data in the group of data has been formatted. The baseline position of the group header can be specified as relative to other format regions on a page. There may be multiple group headers within a formatted document page.

PAGETRAILER This format region specifies a trailer that is to be printed on each new page within a formatted document 355. The baseline position of this format type is normally in the bottom margin, but it can be positioned anywhere on a page and its placement can be relative to the other format regions on a page.

Once the format regions are defined relative to each other on a page within a formatted document 355, each layout identifier that may be associated with the data stream 325 is associated with a format region as shown in step 530. Accordingly, each data record of the data stream 325 will be formatted within the format region that the data record's layout identifier has been associated with.

Once each layout identifier is associated with a format region, layout parameters are defined to control the placement of the associated data records within the format region for each layout identifier in step 540. The user may also specify within the layout parameters certain fixed data to be included in the formatted document 355 every time a certain layout identifier is encountered in the data stream 325.

Once all the layout parameters have been specified for all the layout identifiers applicable to a particular data stream 325, the layout parameters are sent to a formatter as a set of layout parameters 335 as shown in step 550.

It is envisioned that the layout program 330 could be implemented in software or hardware in a number of different embodiments. For instance, the layout program 330 could be resident on a computer system 200 that also contains the formatter 340, or it could reside on another computer system 200 that interacts with the computer system(s) 200 containing the formatter 340 over a network. Alternatively, the layout program 330 could be a routine of the formatter 340. Furthermore, the steps and the order of steps in figure 5 are merely exemplary and as such, it is envisioned that a layout program 330 may have more or less steps than illustrated and that the steps may be performed in a different order than illustrated.

An Exemplary Formatter

Figure 6 is a flow diagram showing the steps performed by an exemplary formatter 340 to create a format document data file 345 to be used by an output device to create the final formatted document 355.

Once the data stream 325 is sent to the formatter 340 and the formatter 340 has access to a set of layout parameters 335 as provided by a layout program 330, the formatter 340 merges fixed data from the set of layout parameters 335 with the variable text contained in the data stream 325, and applies the formatting instructions given in the set of layout parameters 335.

First, the formatter 340 receives data stream 325 from the data extraction program 320 as given in step 610. Next, the formatter 340 determines whether it has a set of layout parameters 335 to apply to the data stream 325. For example, in one embodiment, the formatter 340 may retrieve a set of layout parameters 335 from an associated data storage device, or in another embodiment, the formatter 340 may query a computer system 200 that is running the layout program 330 to have the computer system 200 send the formatter 340 a set of layout parameters 335. Simply stated, the formatter 340 receives a set of layout parameters 335 as indicated by step 620.

In step 630, the formatter 340 identifies and applies the proper parameters for each of the format regions as specified in the set of layout parameters 335. For instance a document may have a page header, a page trailer and one or more group headers and bodies. Typically, each page of a document will have only a single page header and page footer, but may have multiple bodies and group headers. The location of the format regions may be defined as a specific area on a page of a formatted document 355 or the positioning may be relative to another format region. For example, a group header may be positioned relative to a body that precedes it since the size of the body is likely to vary with the amount of data contained within it.

Next in step 640, each layout identifier is associated with a particular format region according to the set of layout parameters 335. Each data record associated with a particular layout identifier is formatted according to the layout parameters associated with the layout identifier in step 650. Step 640, and step 650 are repeated for each data record contained within the data stream 325.

Once all of the data records in the data stream 325 are formatted, the resulting formatted document data file 345 is saved, sent to an output device, or both as shown in step 660.

It is envisioned that the formatter 340 could be implemented in software or hardware in a number of different embodiments. For instance, the formatter 340 could be resident on a computer system 200 that also contains the data extraction program 320 and/or the layout program 330, or it could reside on another computer system 200 that interacts with the computer systems 200 containing the data extraction program 320 and the layout program 330. Alternatively, the formatter 340 could be a routine of the data extraction program 320 or the layout program 330. Furthermore, the steps and the order of steps in figure 6 are merely exemplary and as such, it is envisioned that a formatter 340 may have more or less steps than illustrated and that the steps may be performed in a different order than illustrated.

A Bank Statement Example

Figure 7 is an example of a data stream 700 that might be sent from a data extraction program 320 for a particular checking account. The data stream 700 is comprised of line data records 710 and layout identifiers 705. Each distinct layout identifier identifies the type of line data 710 that follows the layout identifier 705 in a particular line. For instance, the layout identifier 715, “statsum” indicates in the example that any data following it is statement summary data, and the layout identifier 720 “ckdatal” indicates that any data following it is information about a particular check that is to be formatted in a left column in the monthly statement.

Figure 8 is an example of a monthly bank statement 800 formatted using an embodiment of the invention. Area 810 indicates the place on the example statement where the statement summary data associated with the layout identifier 715 “statsum” was formatted. Area 820 indicates the place on the example statement where the statement summary data associated with the layout identifier 720 “ckdatal” was formatted.

Figures 9 and 10 are examples of portions of the set of layout parameters that might be used by a formatter to layout and format portions of the monthly bank statement 800. Figure 9 provides formatting rules, instructions, and fixed text related to the “statsum” layout identifier 715. Figure 10 provides the rules and instructions relating to the placement of check data related to “ckdatal” layout identifier 720. The rules, instructions and fixed text of figure 9 and 10 along with additional rules, instructions, and fixed text for each of the other layout identifiers 705 that appear in the data stream 700 collectively comprise a set of layout parameters for a bank statement.

Figure 11 is a block diagram of a computing environment that might be used by a bank or other financial institution to generate the bank statement 800. Information activity on the bank account is entered into a database of bank transactions 1110 from a variety of sources including, but not limited to, ATM machines 1105, and bank teller terminals 1106. On a periodic basis, such as monthly, a statement data extraction program 1120 extracts all data relating to a particular bank account from the database of bank transactions 1110. The statement data extraction program 1120 uses the data to calculate new pieces of data that will be included in the monthly bank statement 800. For instance, the balance at the beginning and end of the statement period may be calculated. The total dollar amount of credits made to the account and the total dollar amount of checks written against the account may also be determined for statement period. The data is then formatted into a data stream 700 of line data records 710 to which a layout identifier 705 is added to the front of each line data record to indicate the type of information contained within in the line data record relative to the line data record's eventual location in the formatted monthly statement 800. Finally, the statement data extraction program 1120 sends data stream 700 to a statement formatter 1130.

Typically, at some point prior to generation of the data stream 700 by a statement data extraction program 1120, a set of bank statement layout parameters must be generated by a layout program 1140. The layout rules and instructions are generated for each layout identifier that may appear in an associated data stream 700, and they are put into a format that is capable of being read and executed by the statement formatter 1130. A set of bank statement layout parameters 1145 for each of the associated layout identifiers is sent to the statement formatter 1130 over a network connection 1141. As shown in figure 11, the layout program is resident on a computer system 200 separate and distinct to the computer system 200 that holds the database of bank transactions 1110 and the statement data extraction program 1120. Although this separation is not necessary to practice the invention in this example, it may be desirous to prevent the layout program operator from having access to the bank's customers sensitive and personal financial information.

When the statement formatter 1130 receives the data stream 700 from the statement data extraction program 1120, it applies the rules and instructions contained in the set of bank statement layout parameters 1145 to each line data record 710 based on the layout identifier associated with the line data record to format that line data record. For instance, when the formatter 340 encounters a line data record associated with the layout identifier 715 "statsum", it references and applies the layout parameters given in figure 9. The statement formatter 1130 first identifies that information associated with the layout identifier 715 "statsum" is to be placed within the "Body" format region on a page as shown by instruction 905. The fixed (or static text) term, "Super Checking Account Activity" is formatted at the specified position (0" in from the left margin and 2" down from the top of the "Body" format region) in the "super" font, then three bold lines are placed in the specified position as per instruction 910. Various headers of fixed text are placed at various positions from the left margin, 2.4" down from the top of the "Body" as per instruction 915. Next, according to instruction 920, the various pieces of variable text in the form of dollar amounts are placed below the headers. For instance, instruction 920 directs the statement formatter 1130 to place the text, $2591.24, which begins at the 11th byte of information in the associated line data record, 1.4" in from the left margin and 2.6" down form the top of the "Body." Finally, a fixed line is placed below the variable text as per instruction 925.

Additionally, when the statement formatter 1130 encounters a line data record associated with the layout identifier 720 "ckdatal", it references and applies the layout parameters given in figure 10 to the line data record. The statement formatter 1130 first identifies that the line data record associated with the layout identifier 720 is to be placed within the "Body" format area on a page as shown by instruction 1005. As per instruction 1010, the check number from the line data record is placed 1.4" in from the left margin of the "Body" format region in the next available line for formatting relative to the last previously formatted line. Next, as per instructions 1015 and 1020, the date and the amount of the check are placed on the same, or current, line as the check number in from the left margin of the "Body" 2.4" and 4.2" respectively. When a statement formatter 1130 reads additional line data records associated with the "ckdatal" layout identifier 720, it formats the next additional line data record in the next available print line within the "Body" format region. For instance in figure 8, area 820, check number 353 information is placed below the information for check 352, and the information for check 354 is placed below check 353.

Once the statement formatter 1130 formats all of the statement line data records for a particular bank statement, the file containing the formatted document 800 is sent to a printer 1150 for printing.

Although the example of a bank statement is presented, the content of a document formatted using this invention is limitless and may encompass, but is not limited to, financial statements, telephone bills, utility bills, invoices, and inventory statements. The bank statement example is, therefore, merely, illustrative.